

Seminarios 4U

Diseño de autoevolución y autoadaptación en tiempo de ejecución para sistemas de software inteligentes

Miguel Jiménez^{1,2}, **Luis F. Rivera**^{1,2}, Norha M. Villegas¹, Gabriel Tamura¹, Hausi Müller¹
miguel@uvic.ca, rivera@uvic.ca

¹ Rigi Research, University of Victoria, British Columbia, Canadá

² i2t-DRISO, Universidad Icesi, Cali, Colombia

Organizadores



Nuestro Equipo

Context Management



Norha Villegas, Ph.D.

Associate Professor, Universidad Icesi, Colombia
Dir. of the Soft. Sys. Eng. bachelor program, Universidad Icesi
Adjunct Professor, University of Victoria, BC

Adaptive Systems



Hausi Müller, Ph.D., Peng, FCAE

Professor, University of Victoria, BC
Initiative Co-Chair, IEEE Future Directions Quantum Initiative
Adjunct Professor, Universidad Icesi, Colombia

Runtime Verification & Validation



Gabriel Tamura, Ph.D.

Full Professor, Universidad Icesi, Colombia
Head of the i2t/DRISO Research Group
Adjunct Professor, University of Victoria, BC

Digital Twins



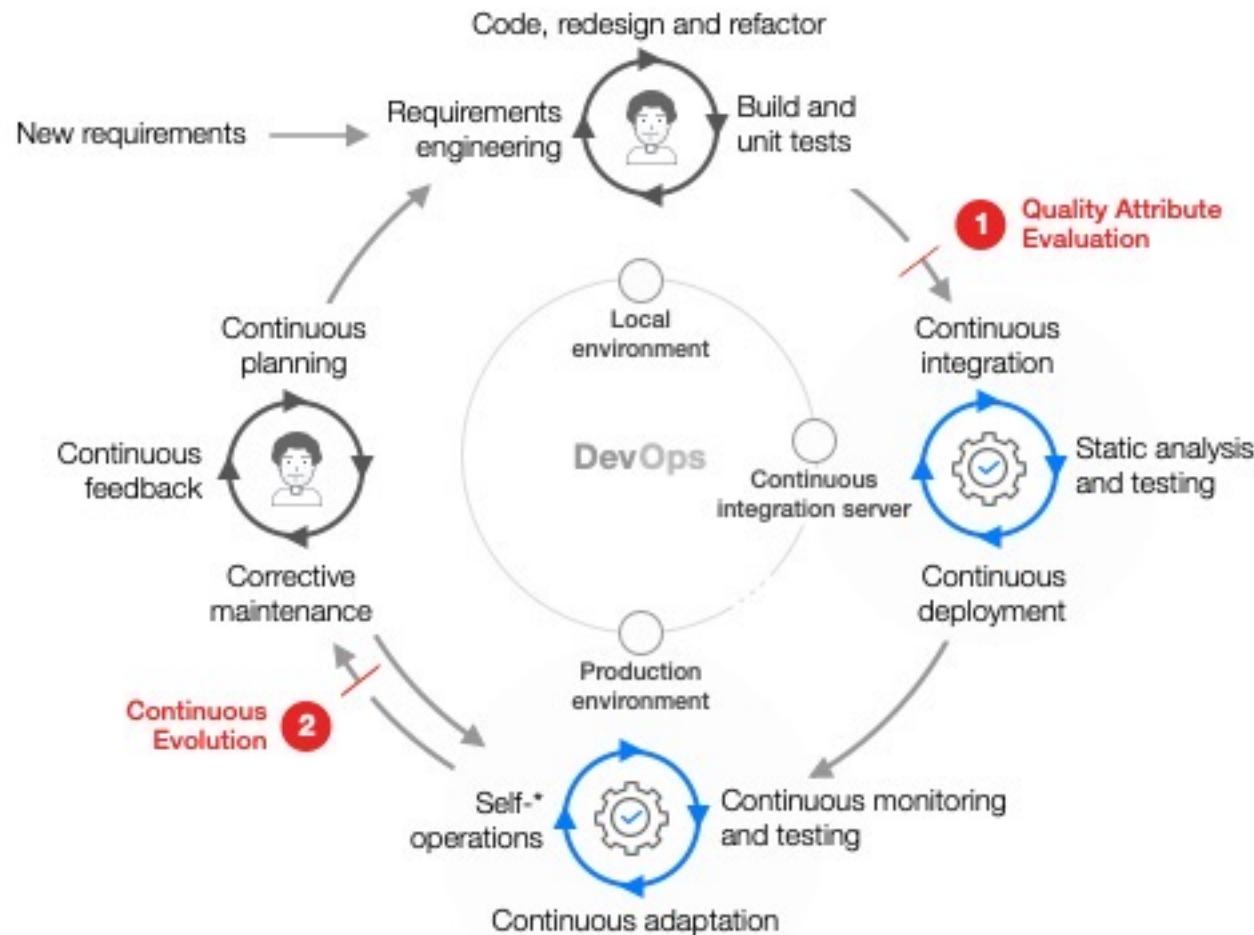
Felipe Rivera, Ph.D. Student
University of Victoria, BC

Continuous Experimentation



Miguel Jiménez, Ph.D. Student
University of Victoria, BC

Discontinuidades en el Proceso de Entrega



- La evolución en el corto plazo está bien desarrollada
- Hay mucho por explorar respecto a evolución en el largo plazo
- La evaluación de atributos de calidad no está bien integrada
- AIOps se concentra en análisis de causa raíz

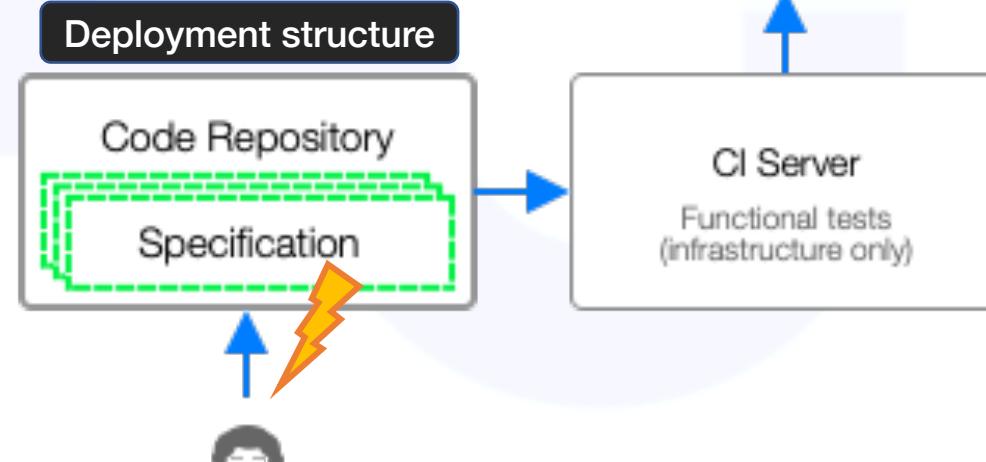
Caso de estudio:

- IBM CAM

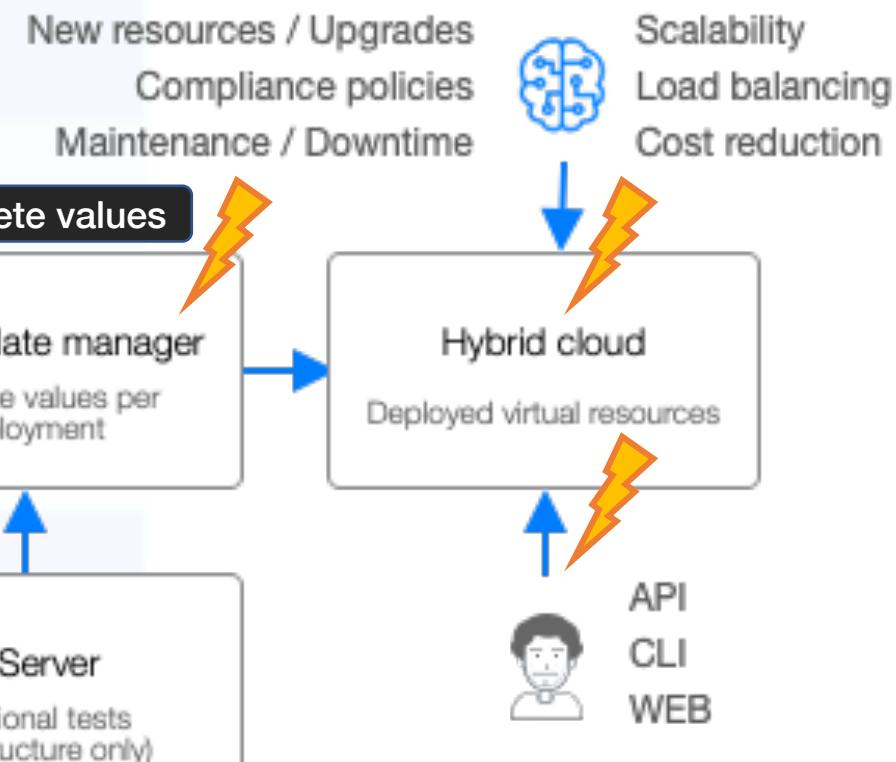
Ejemplo No. 1: Infraestructura como código (IaC)

- Así es como debería lucir un proceso continuo
- Enfoques sistemáticos para mantener la correspondencia entre diseño y código son raramente usados
- Una integración limitada restringe el impacto de componentes autónomos

IaC espera cambios aquí



Pero estos también pasan aquí



¡Y aquí!

Caso de estudio:

- IBM CAM

Ejemplo No. 1: Infraestructura como código (IaC)

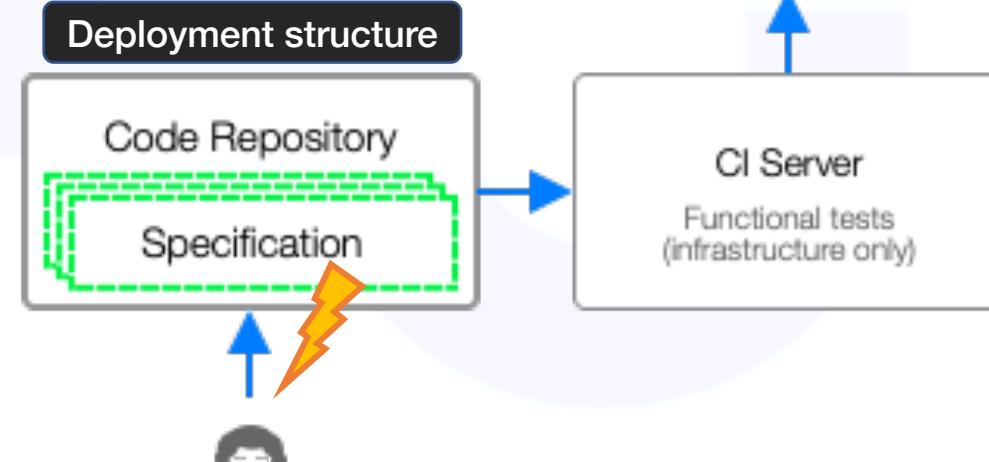
Adopción de IaC

- Miles de recursos desplegados
- Creación manual es propensa a errores
- Es un proceso costoso

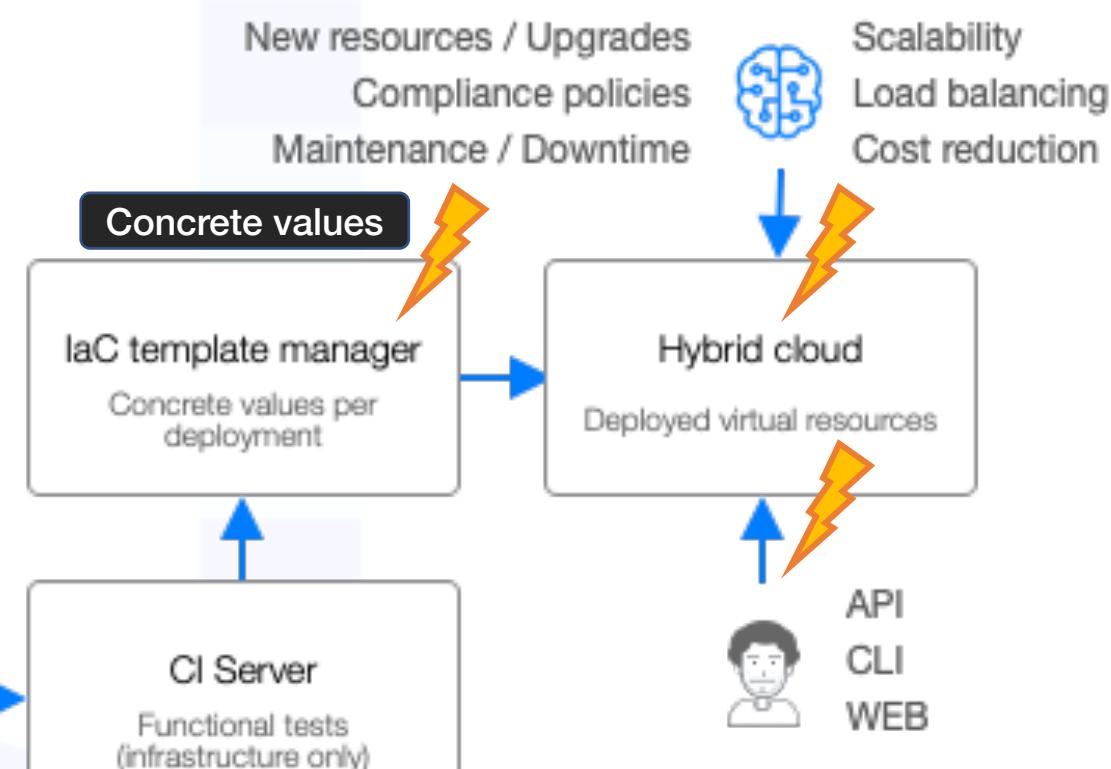
Evolución de las especificaciones

- Desechar automatización existente
- Las inconsistencias llevan a la deuda técnica

IaC espera cambios aquí



Pero estos también pasan aquí



¡Y aquí!

Casos de estudio:

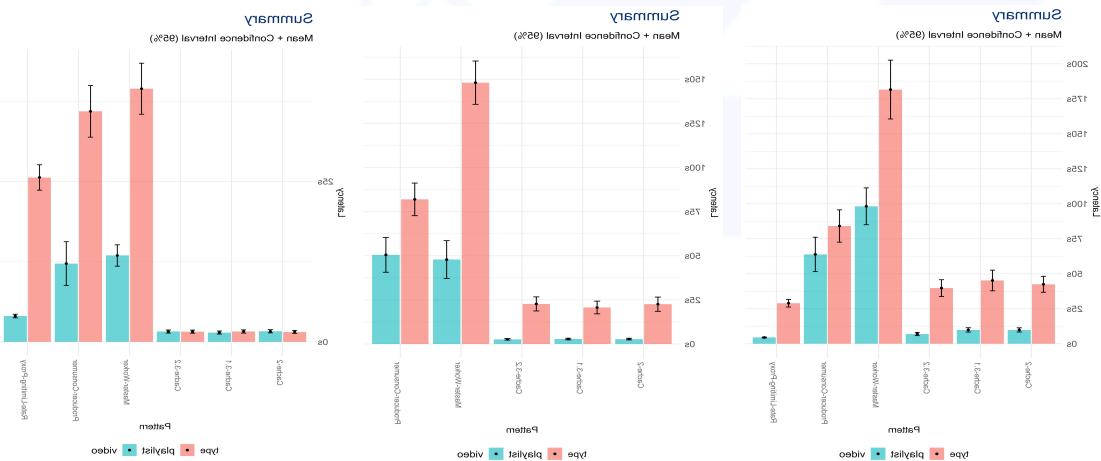
- MIO – Cali
- Cloud systems

Ejemplo No. 2: Evaluación de Patrones Software

Configuración automática del clúster



La arquitectura software como una función del ambiente



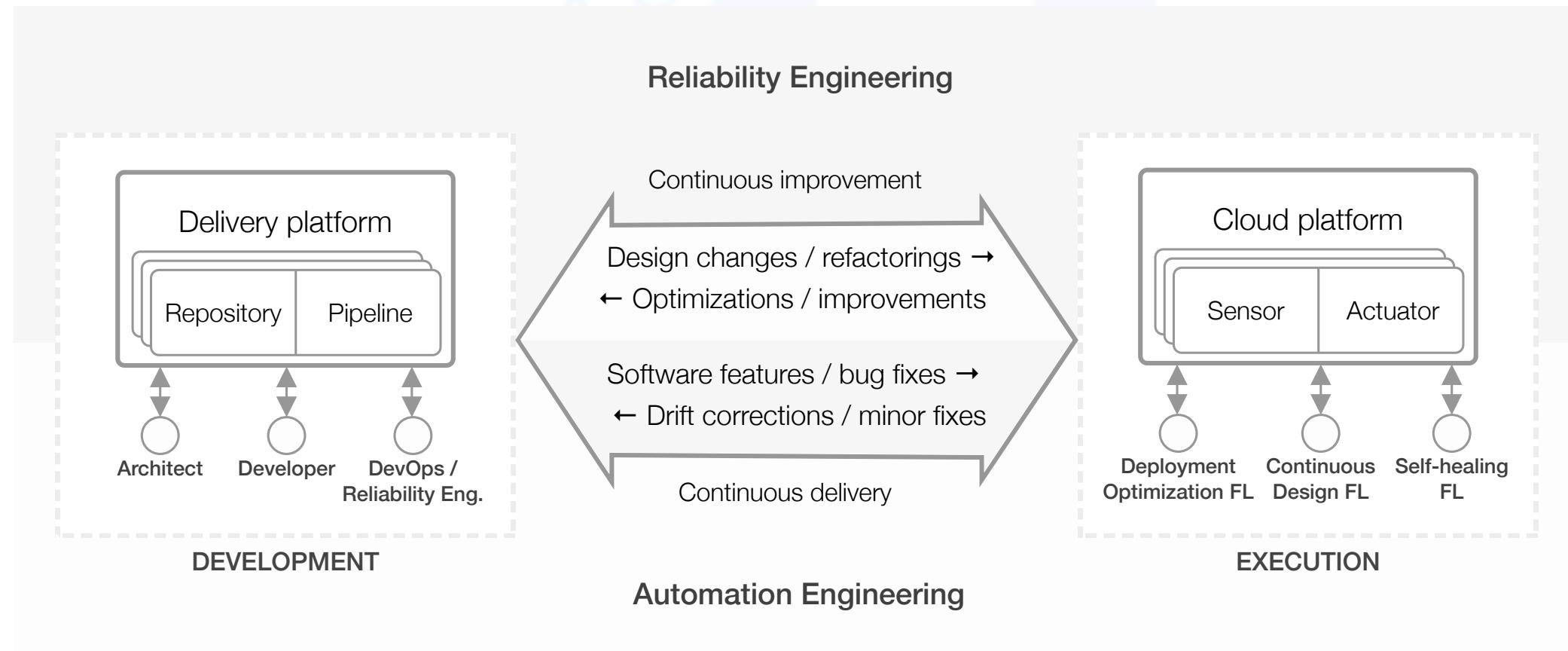
- Evaluación cuantitativa de los patrones software (incluyendo despliegue)
- Las descripciones son suficiente para la ingeniería del software?
- Cómo encaja la evaluación de atributos de calidad en la entrega continua?
- La falta de integración restringe una evolución más avanzada en tiempo de ejecución

Retos para la especificación y el manejo del despliegue

1. Soporte en herramientas, y al nivel de la notación, para enlazar conceptos de despliegue de tiempo de diseño y ejecución
2. Soporte asistido con herramientas para la evolución de especificaciones de despliegue y manejo de la configuración en tiempo de ejecución

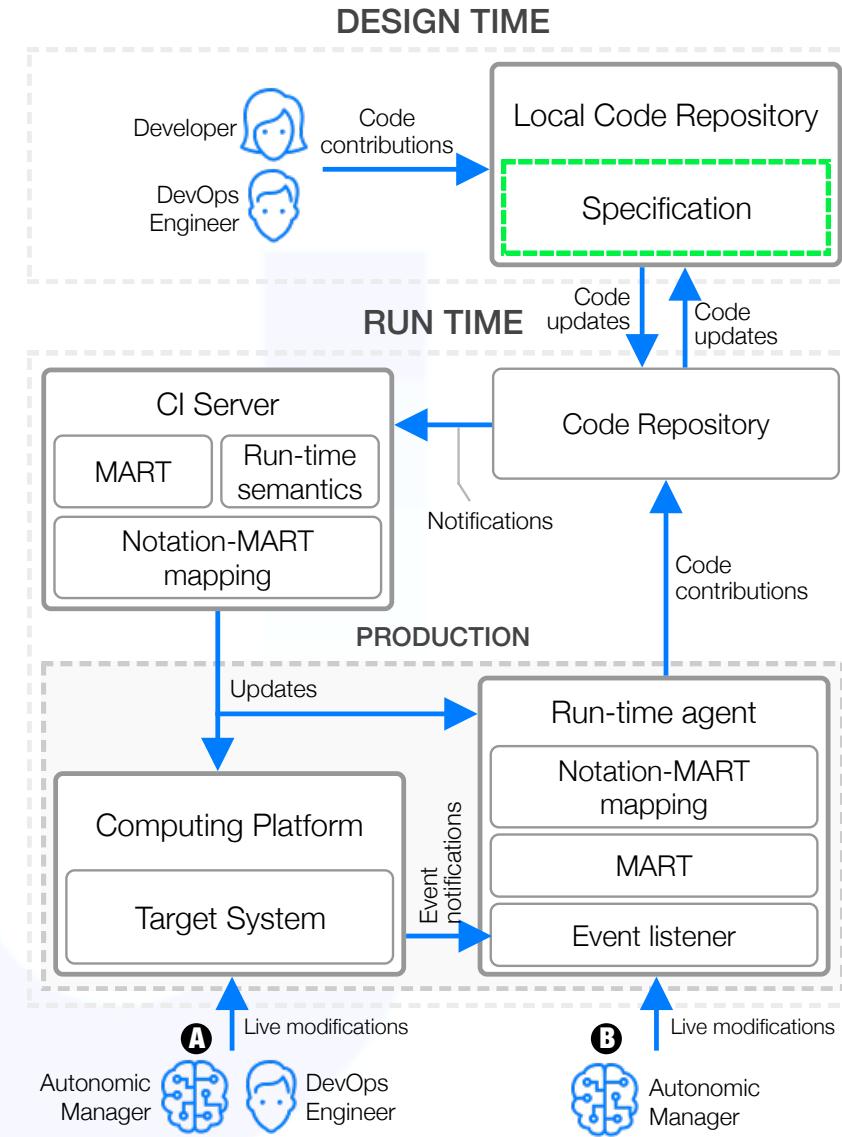
Entrega continua bidireccional

- Ingeniería de automatización vs confiabilidad
- Entrega continua vs mejoramiento continuo
- Ciclos de retroalimentación independientes y coexistentes



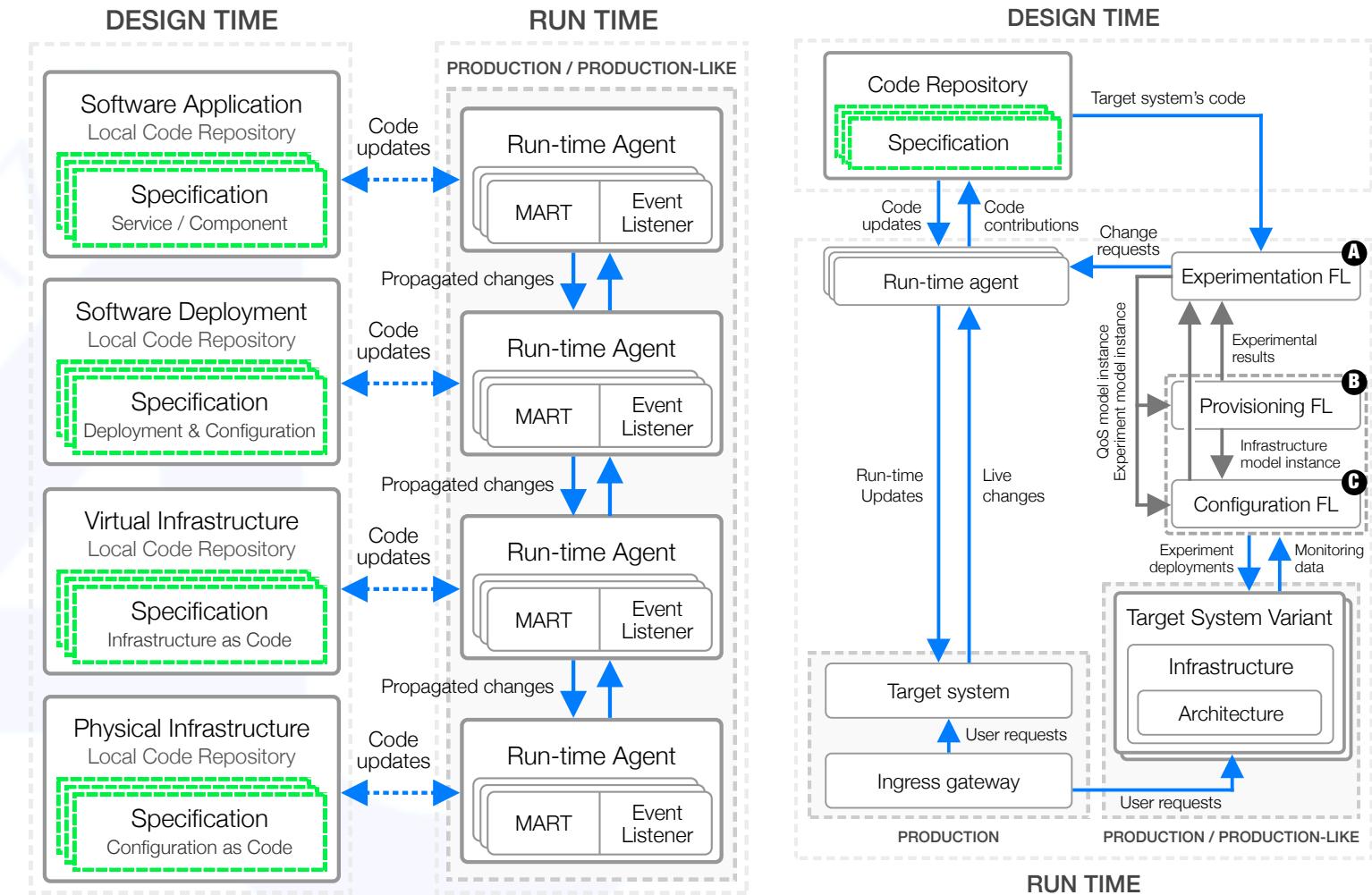
Integración continua bidireccional

- Un agente relaciona modelos en tiempo de ejecución con cambios en el código
- Línea base para la evolución en el largo plazo



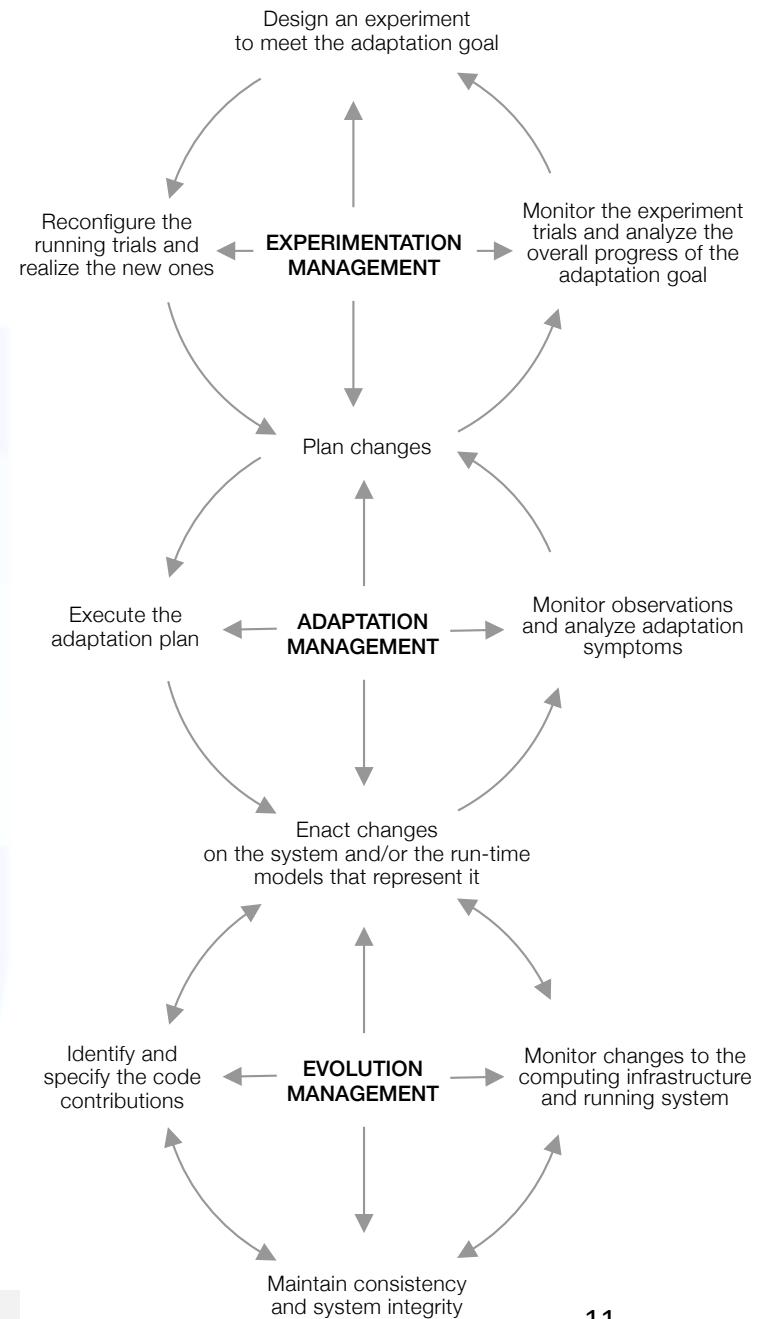
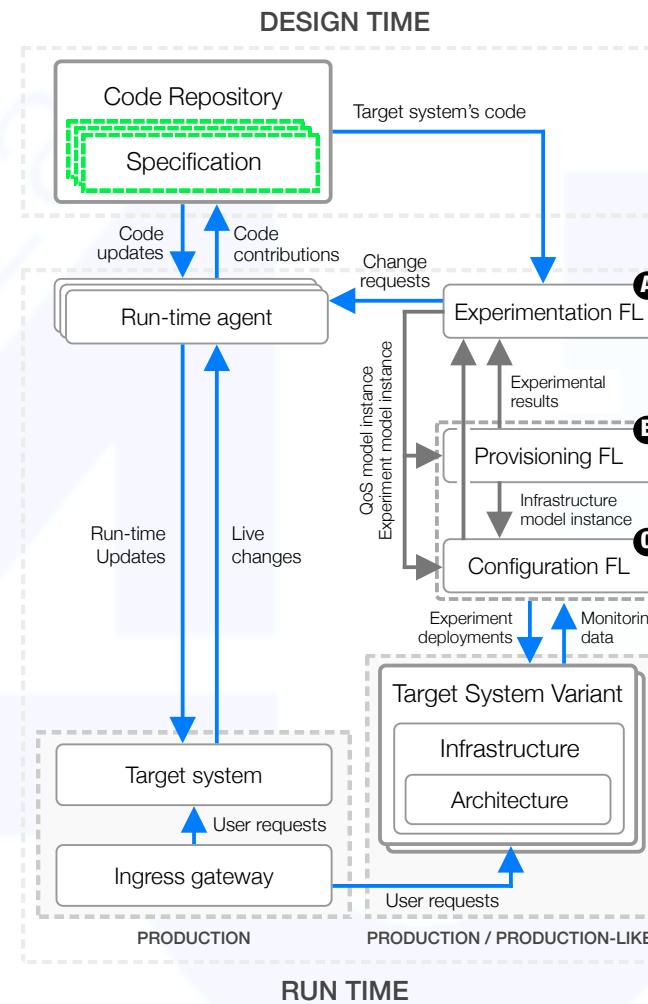
Experimentación dirigida por la calidad

- Red de modelos en tiempo de ejecución que representan preocupaciones transversales
- Aún hace falta la integración con el desarrollo
- Se acerca más a nuestra visión en el largo plazo pero hace falta una definición más genérica



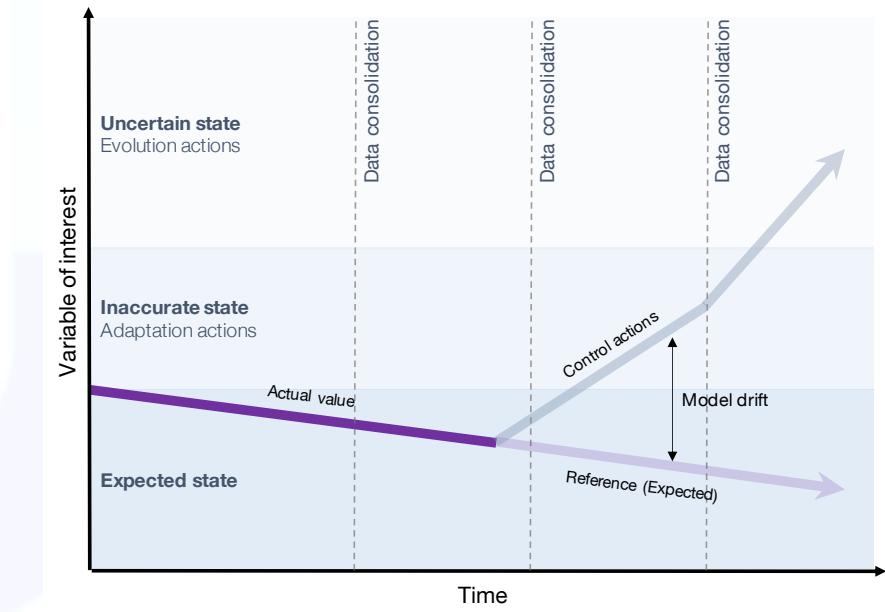
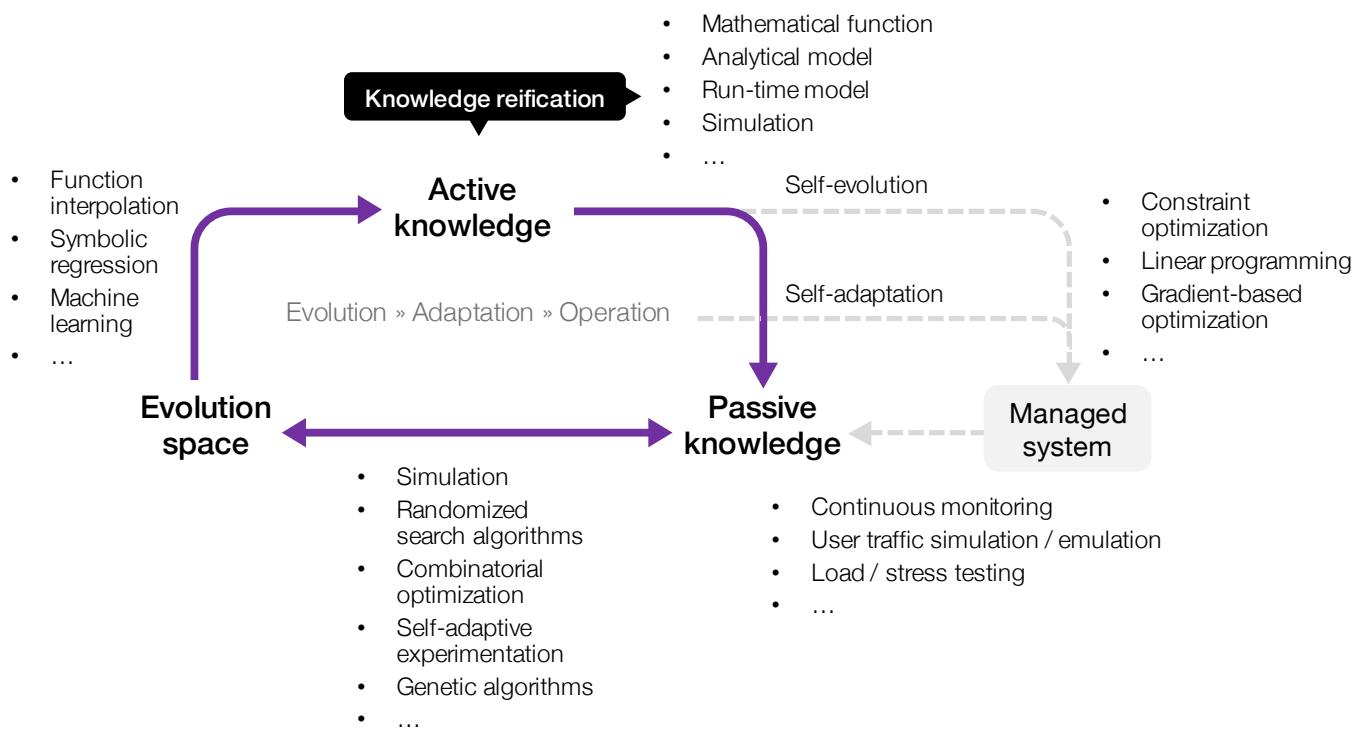
Experimentación dirigida por la calidad

- Red de modelos en tiempo de ejecución que representan preocupaciones transversales
- Aún hace falta la integración con el desarrollo
- Se acerca más a nuestra visión en el largo plazo pero hace falta una definición más genérica



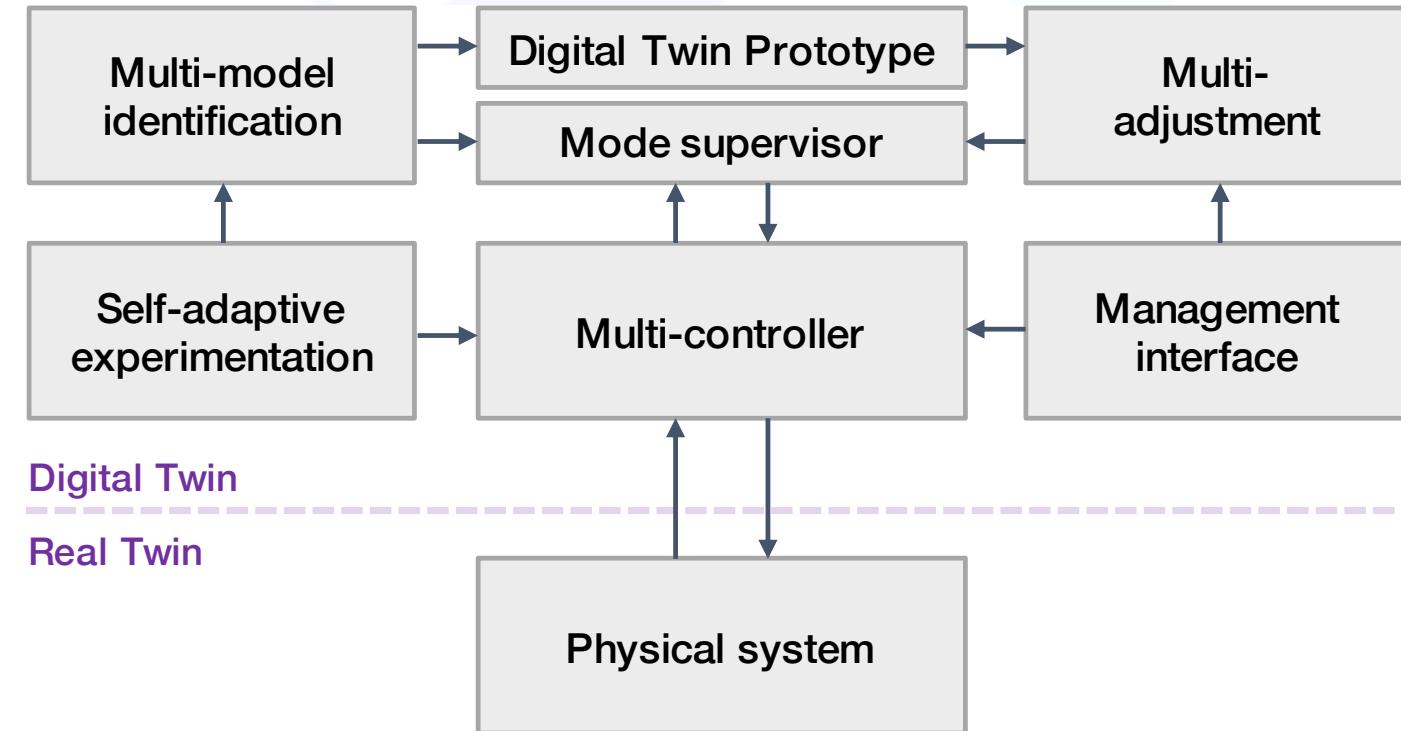
Hacia la definición de un proceso de evolución

- Proponemos una relación explícita entre evolución, adaptación y operación
- Conocimiento pasivo vs activo
- Técnicas para cosificación de conocimiento

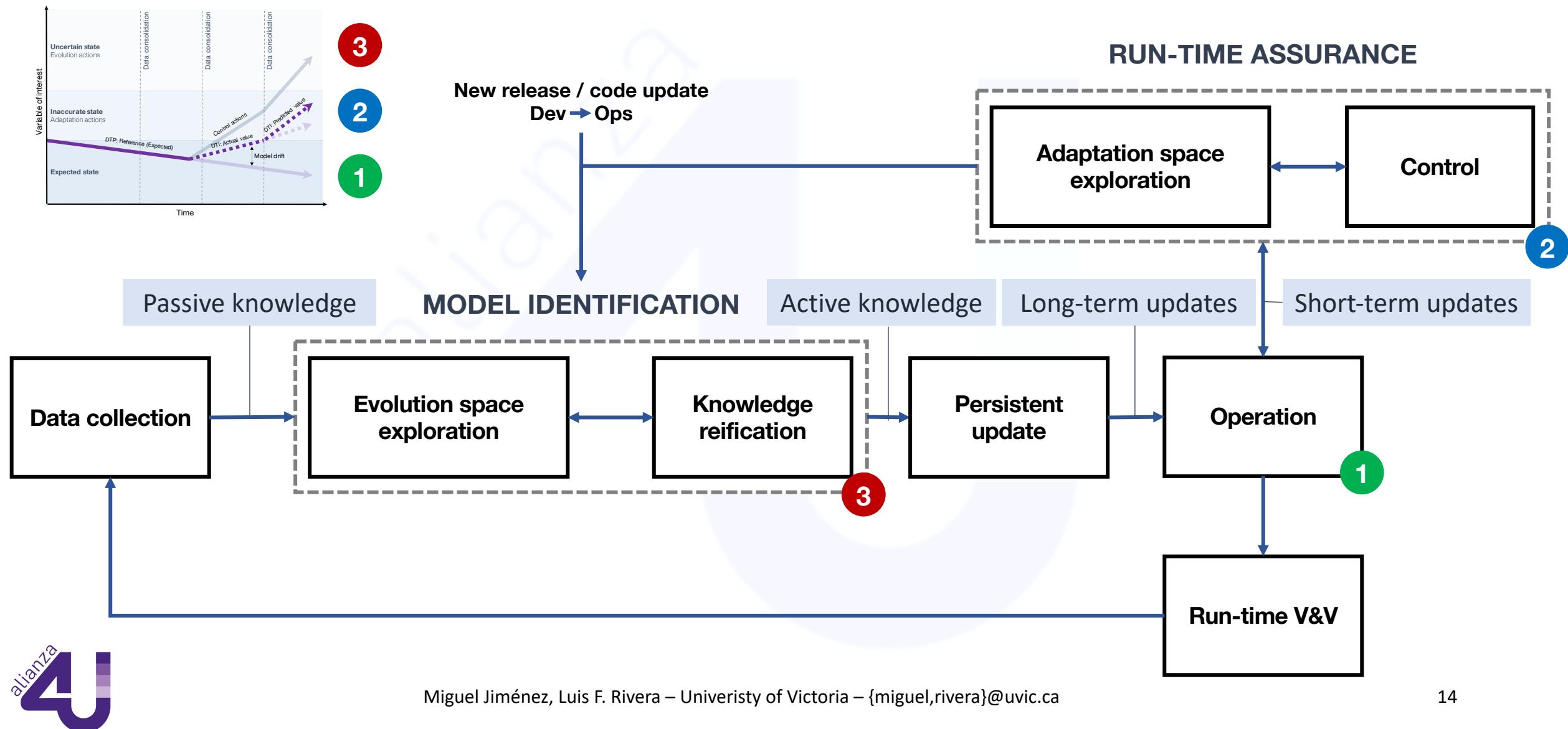


Una arquitectura de referencia

- Autonomía confiable
- Resiliencia operacional
- Dualidad entre evolución y adaptación (Corto vs largo plazo)
- Originalmente propuesta para sistemas ciberfísicos

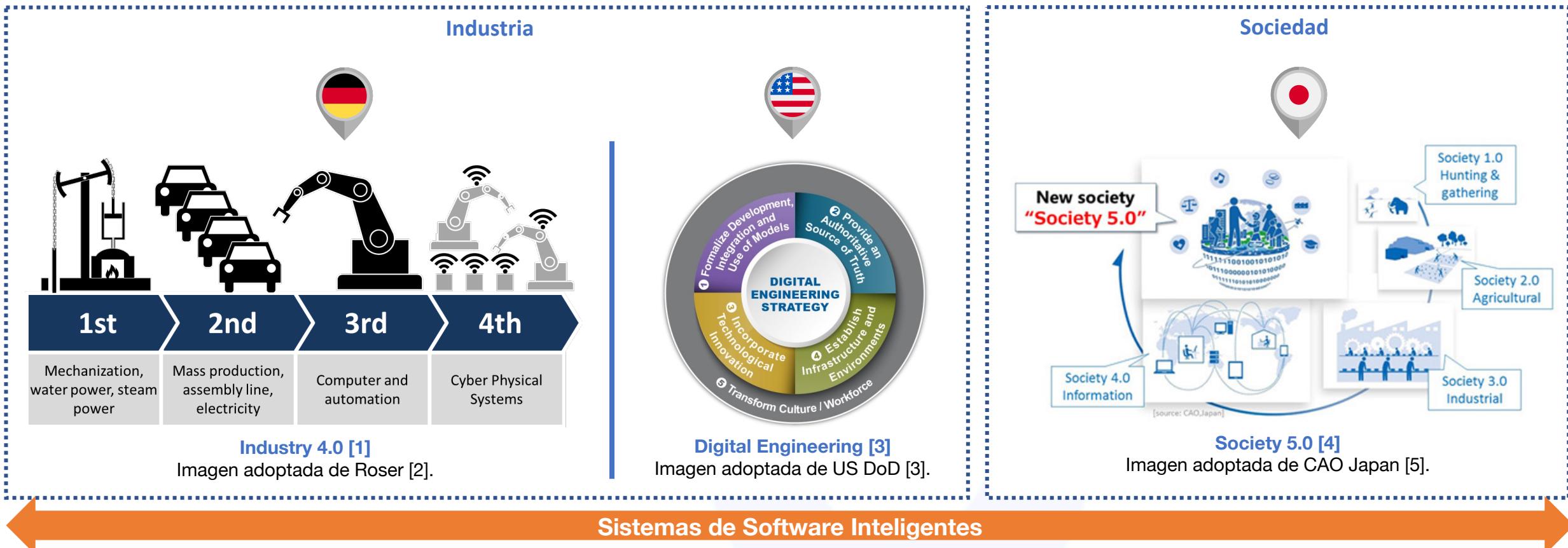


Un proceso de evolución en tiempo de ejecución (trabajo en progreso)





Dualidad entre desarrollo y operaciones más allá de la Ingeniería de Software



[1] H. Lasi, P. Fettke, H.G. Kemper, T. Feld, and M. Hoffmann. Industry 4.0. Business & Information Systems Engineering. 2014.

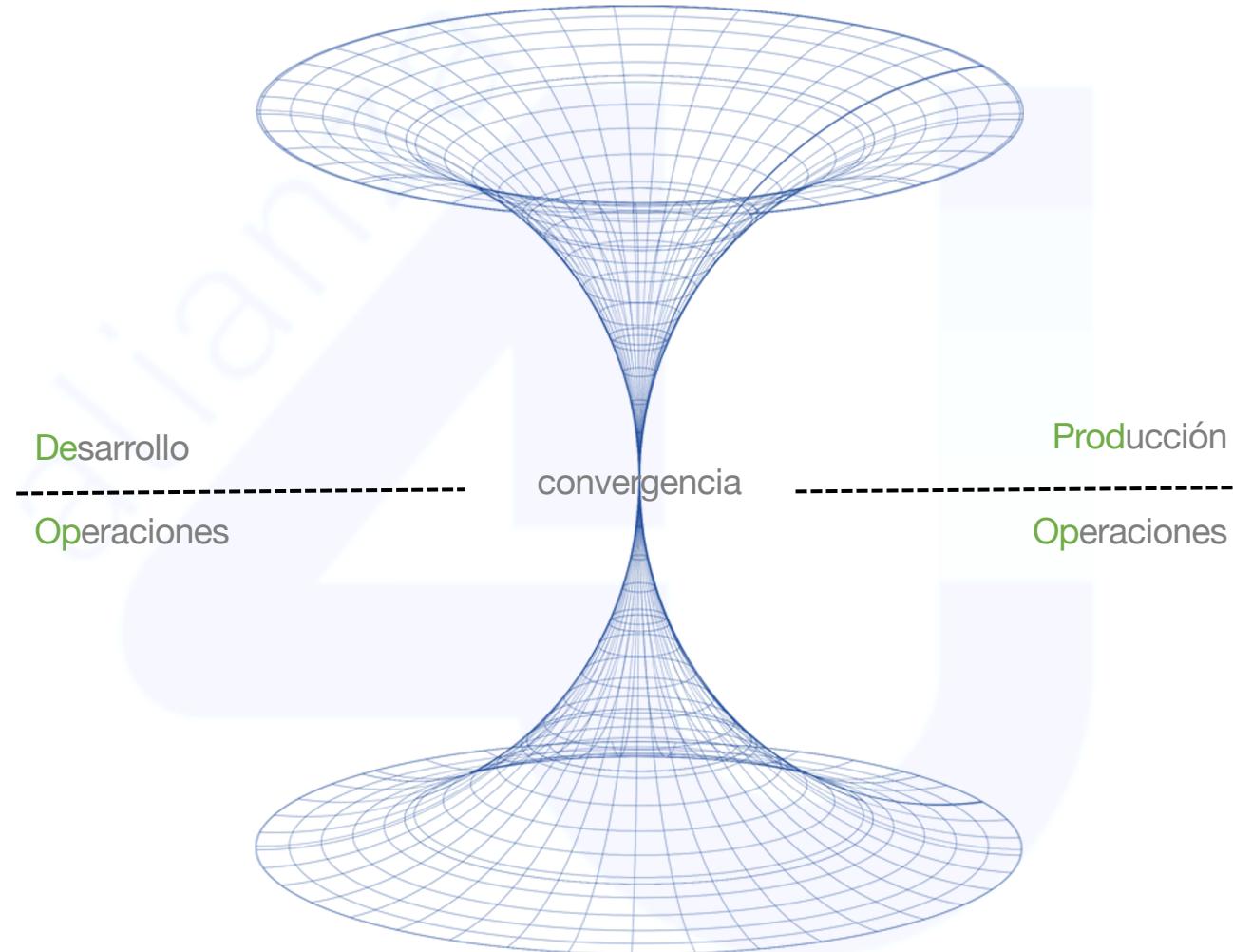
[2] C. Roser. Illustration of Industry 4.0. www.allaboutlean.com. 2016.

[3] USA Department of Defense (DoD). Department of Defense - Digital Engineering Strategy. 2018.

[4] Cabinet Office. The 5th Science and Technology Basic Plan. Government of Japan. 2016.

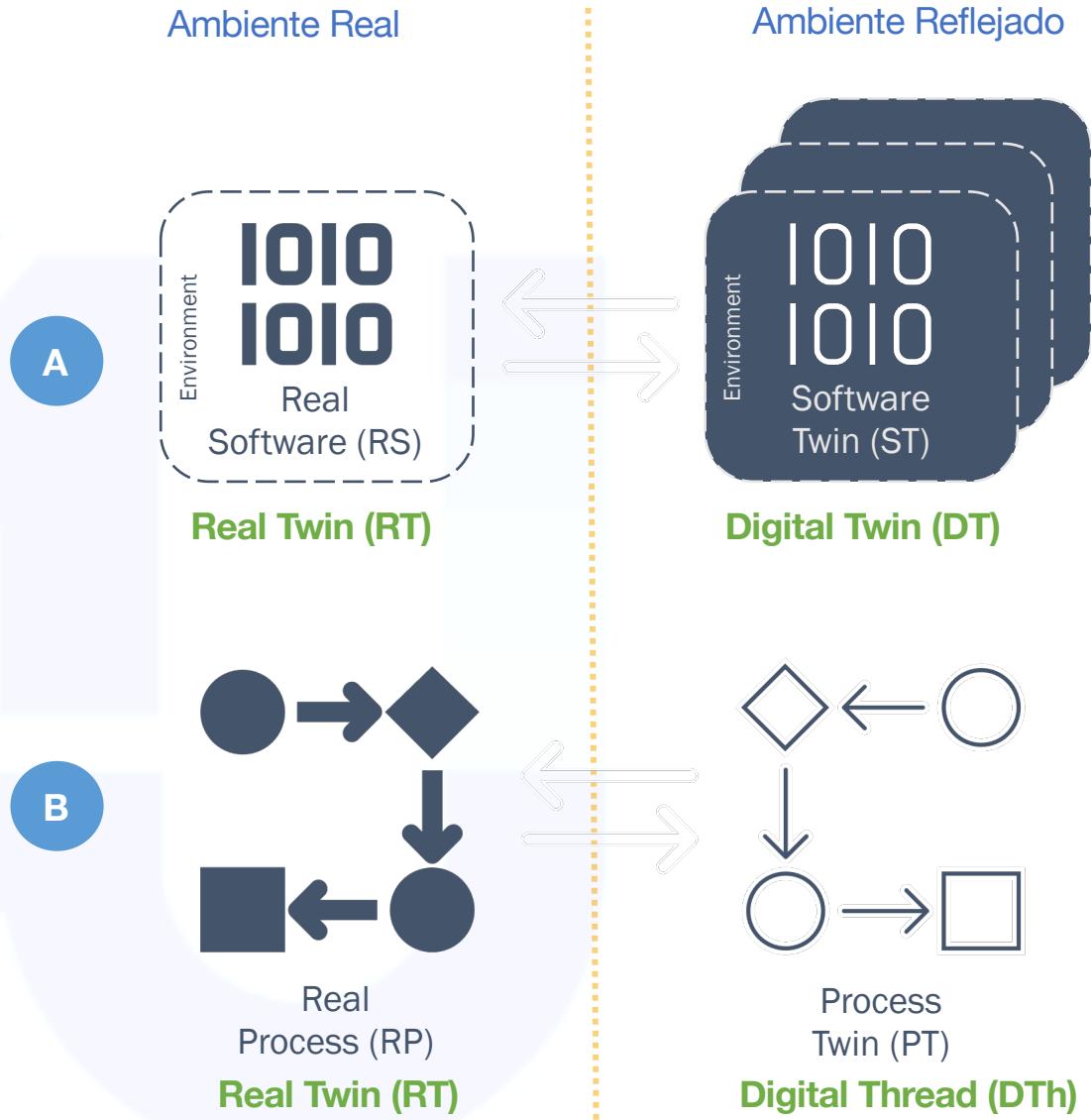
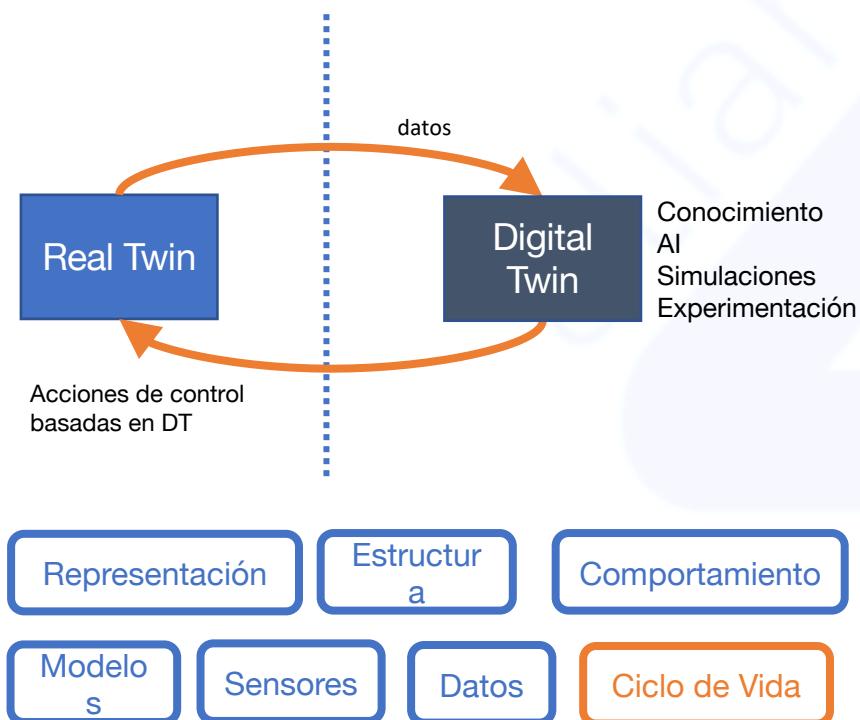
[5] Cabinet Office. Society 5.0. www8.cao.go.jp/cstp/english/society5_0/index.html. 2020.

Dualidad entre Desarrollo y Operaciones más allá de la Ingeniería de Software: Convergencia



Digital Twin (DT)

- Representación virtual de una entidad conceptual o física a lo largo de su ciclo de vida.



Beneficios y tendencias relacionados con la implementación de DTs



Technology

Digital Twin Genie Case Study: 54% reduction in automotive manufacturing costs

businesswire A BERKSHIRE HATHAWAY COMPANY

HOME SERVICES NEWS EDUCATION ABOUT US

Search

Key Takeaways

Specialty

Digital Twins Market, 2020: Up to 89% of all IoT Platforms will Contain Some Form of

Exhibit

Digital Twinning Capability by 2025 - ResearchAndMarkets.com

Exhibit

April 03, 2020 05:07 AM Eastern Daylight Time

STAMFORD, Conn., February 20, 2019

DUBL

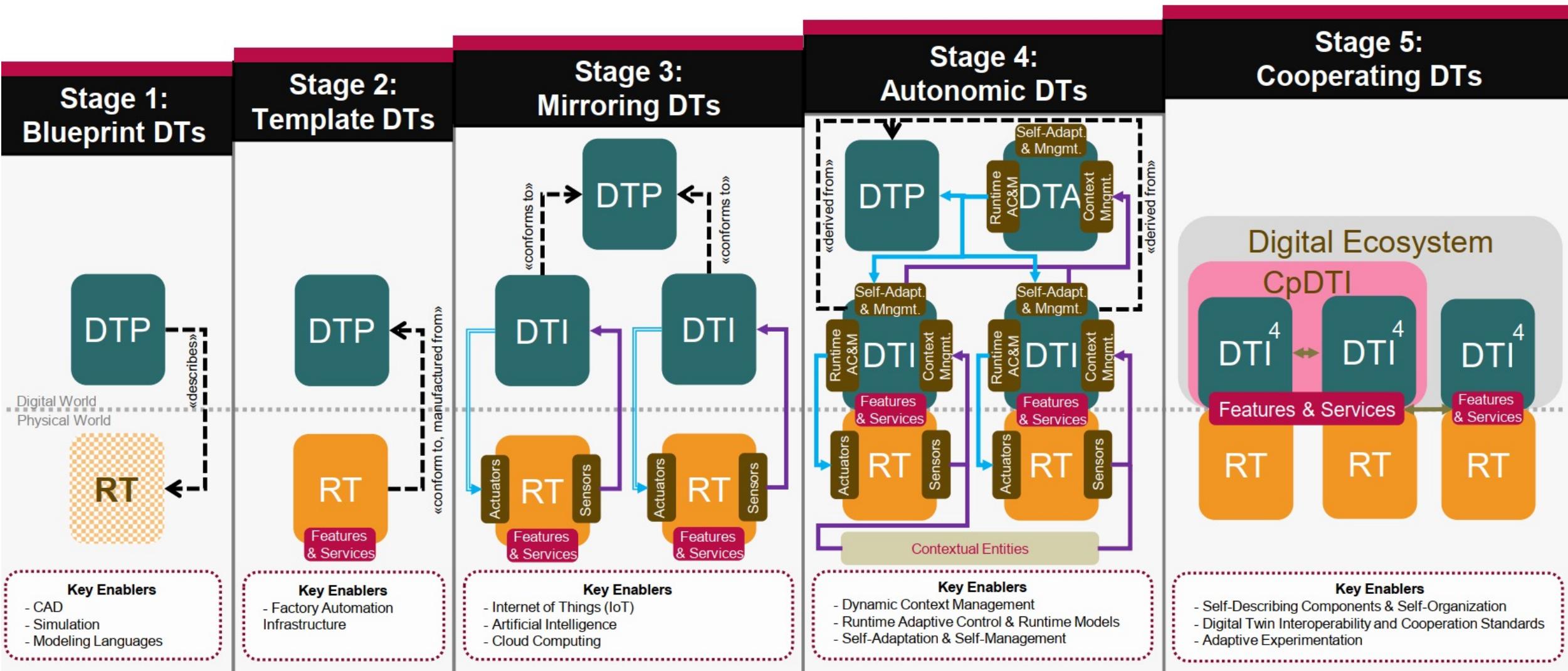
has b

Gartner Survey Reveals Digital Twins Are Entering Mainstream Use

75 Percent of Organizations Implementing IoT Already Use Digital Twins or Plan to Within a Year

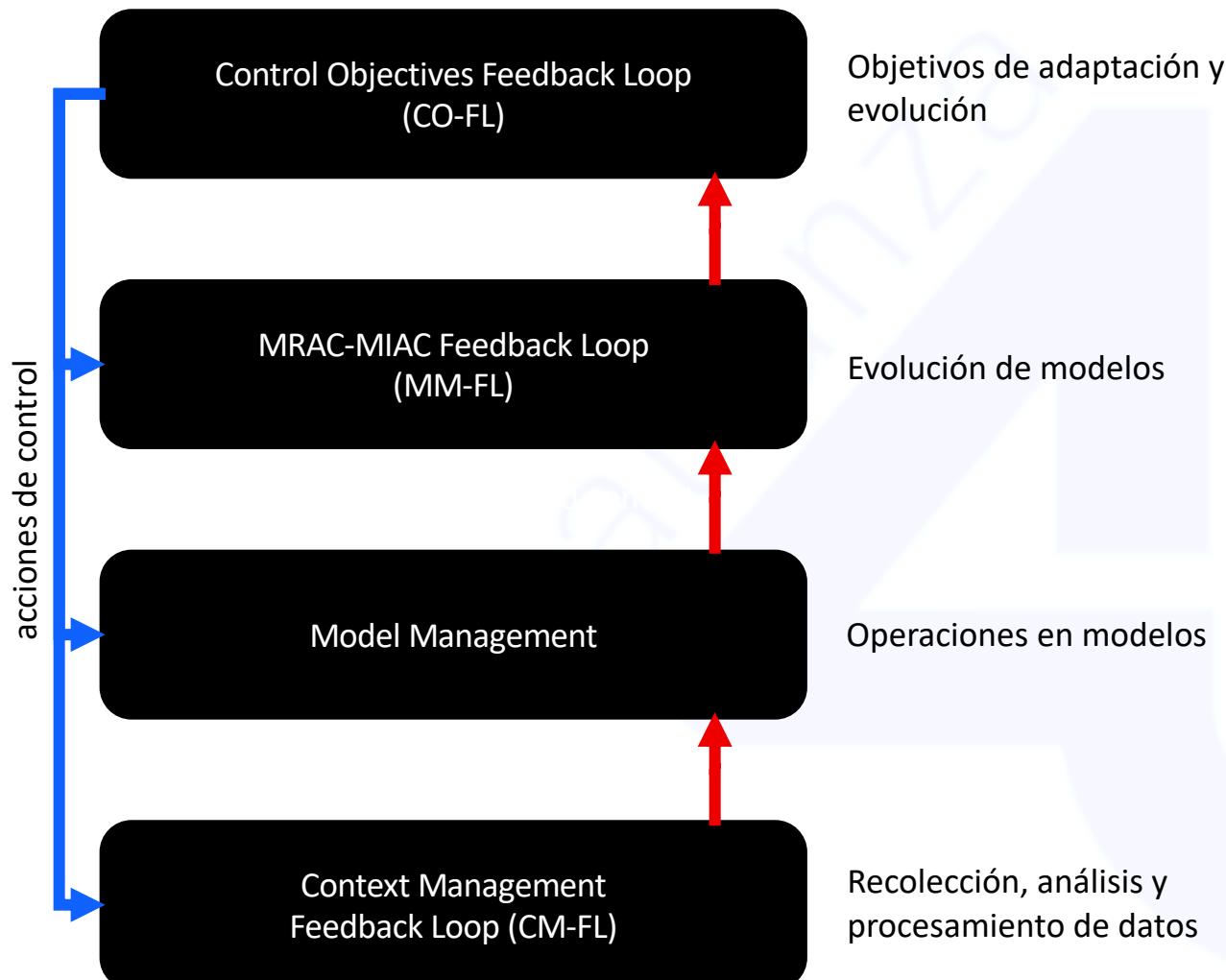
Thirteen percent of organizations implementing [Internet of Things \(IoT\)](#) projects already use [digital twins](#), while 62 percent are either in the process of establishing digital twin use or plan to do so, according to a recent IoT implementation survey* by Gartner, Inc.

Evolución de DTs en la Historia: Modelo de Madurez



L.F. Rivera et al., "The Forging of Autonomic and Cooperating Digital Twins," Internet Computing, In press, 2021.

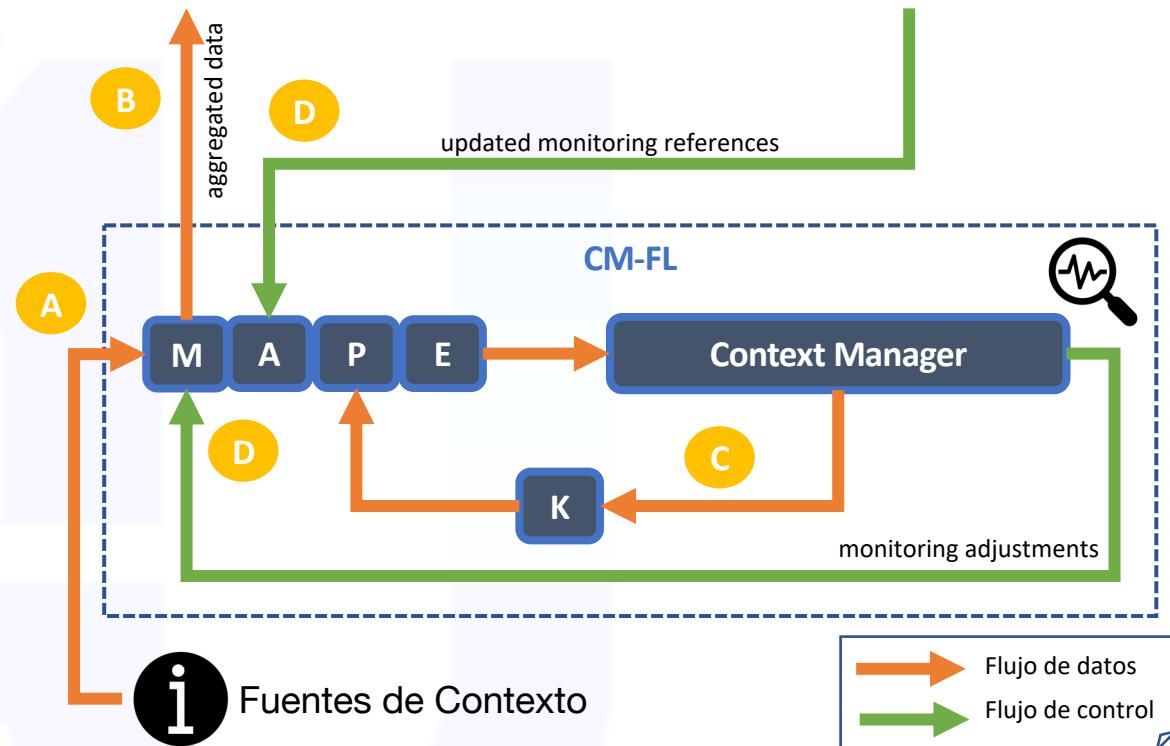
GEMINIS: Modelo de Referencia para Sistema de Software Inteligentes (DTs)



GEMINIS – The Context Management Feedback Loop (CM-FL)

Responsabilidades principales:

- A** Adquisición y manejo de datos contextuales
- B** Agregación de datos
- C** Detección, consolidación y análisis de síntomas de contexto
- D** Adaptación de esquemas de monitoreo

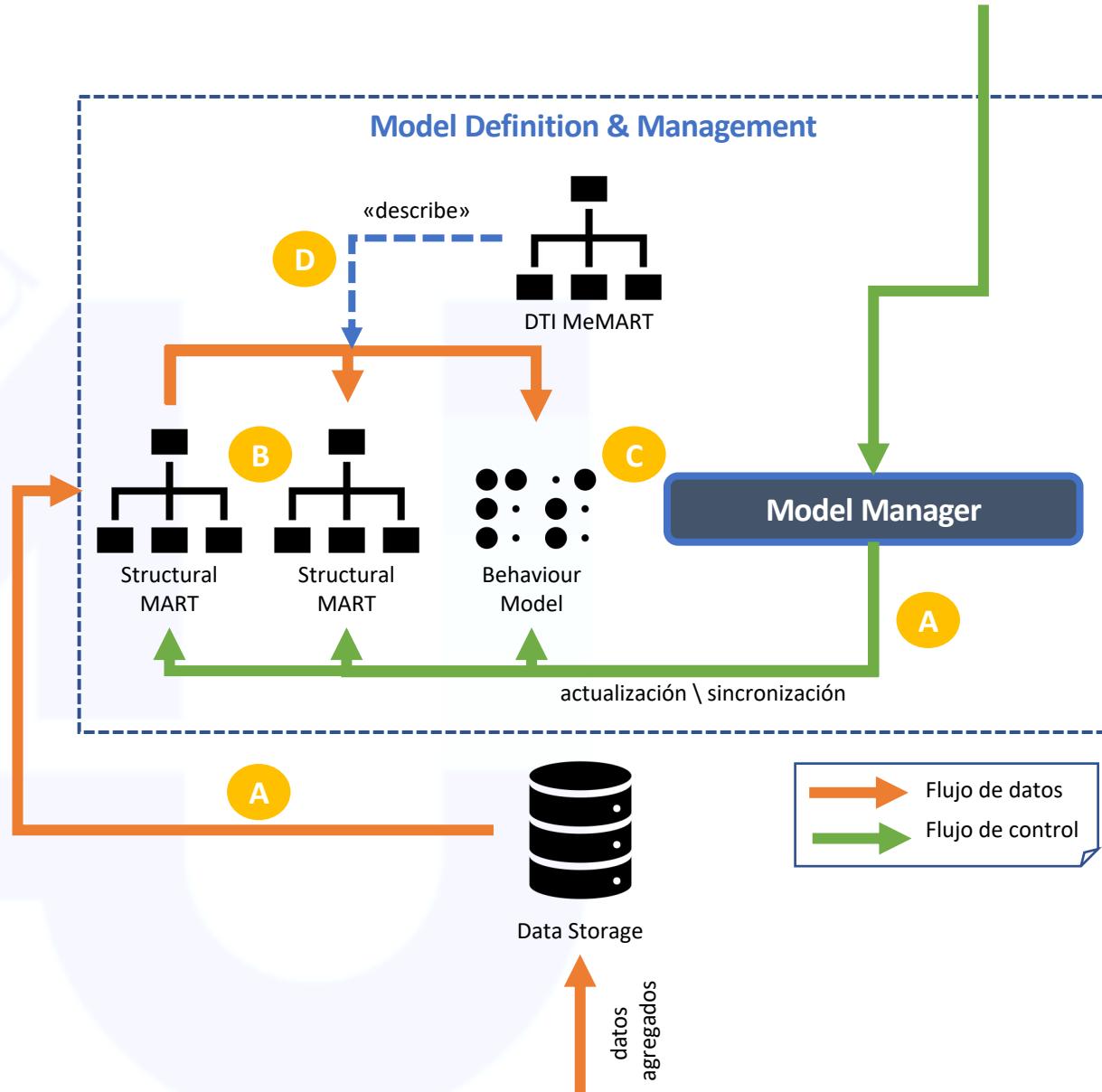


L.F. Rivera et al., *On the Engineering of IoT-Intensive Digital Twin Software Systems*. SERP4IoT (ICSE-Workshops). 2020

GEMINIS – Model Definition and Management

Responsabilidades principales:

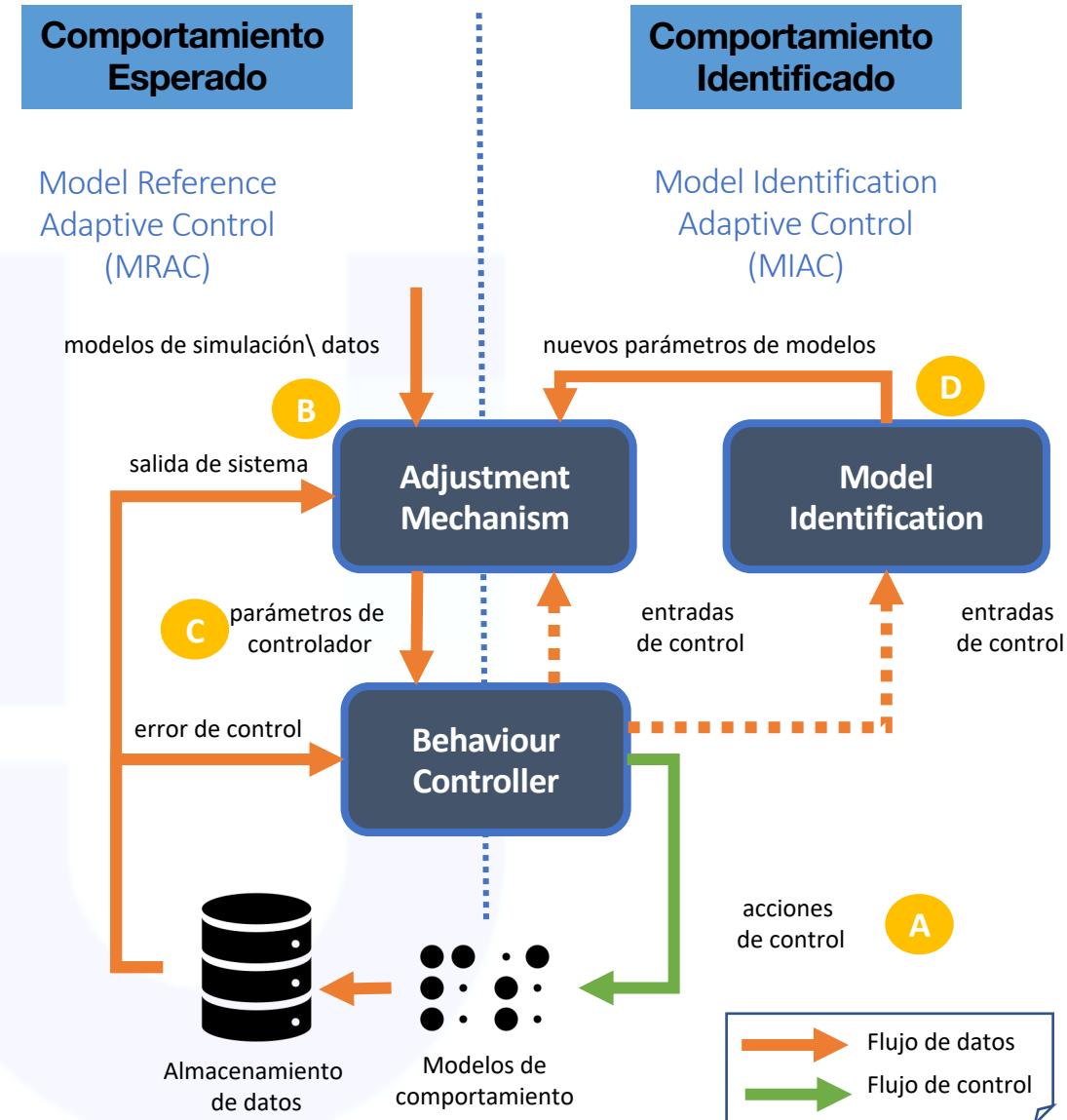
- A** Operaciones y sincronización de modelos
 - B** Estado actual de características representadas
 - C** Descripciones de comportamiento y estructura a través de modelos
 - D** Definición y mantenimiento de relaciones entre modelos



GEMINIS – The MRAC\MIAC Feedback Loop (MM-FL)

Responsabilidades principales:

- A** Evolución continua de modelos de DT
- B** Análisis de comportamiento esperado y actual
- C** Detección de desviaciones en comportamiento esperado y acciones de control necesarias
- D** Refinamiento de modelos basado en técnicas de identificación

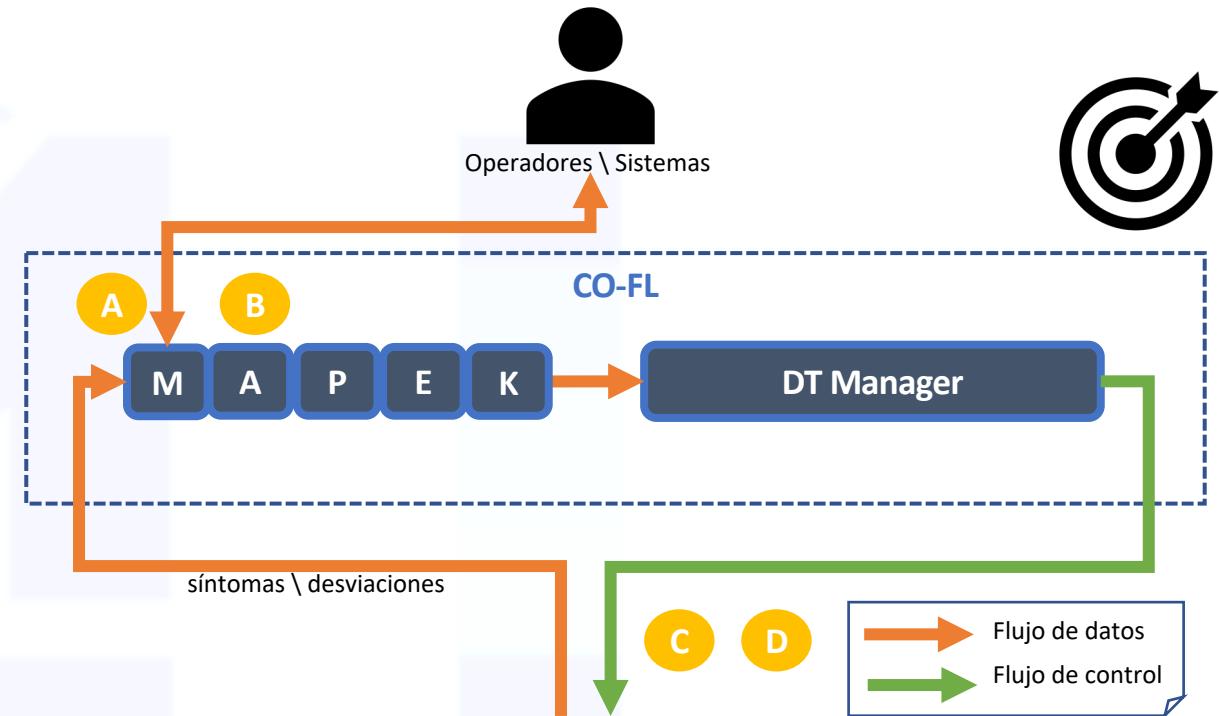


L.F. Rivera et al., *On the Engineering of IoT-Intensive Digital Twin Software Systems*. SERP4IoT (ICSE-Workshops). 2020

GEMINIS – The Control Objectives Feedback Loop (CO-FL)

Responsabilidades principales:

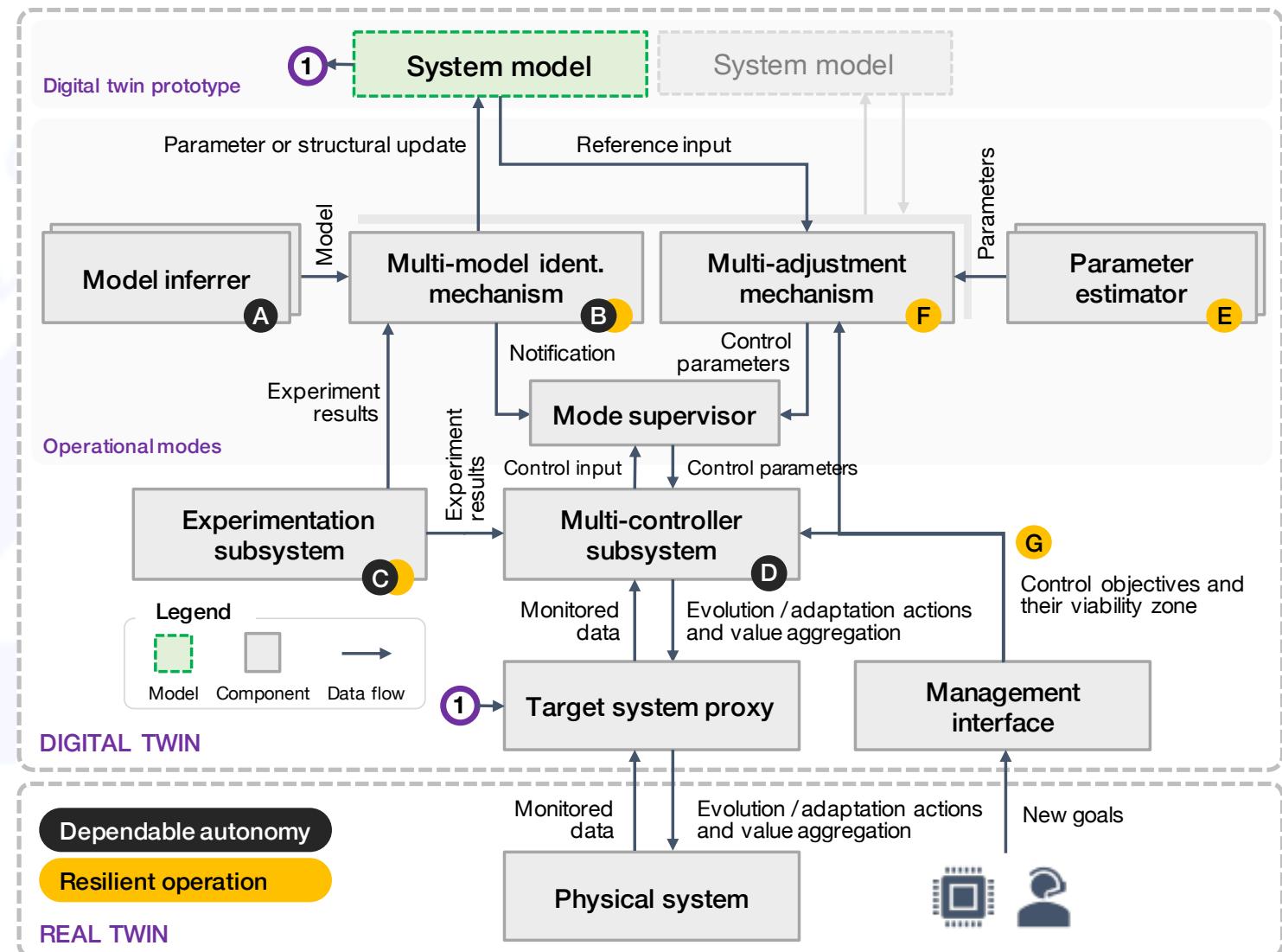
- A** Satisfacción de objetivos
- B** Análisis de condiciones críticas de entidades representadas
- C** Orquestación de evolución
- D** Instancias de Digital Twins



L.F. Rivera et al., **On the Engineering of IoT-Intensive Digital Twin Software Systems**. SERP4IoT (ICSE-Workshops). 2020

Sinergia entre adaptación y evolución en DTs

- Arquitectura de referencia que enfatiza la dualidad \ sinergia entre adaptación y evolución de forma concreta
 - Consolidación de conocimiento activo
 - Integración del concepto de evolución con conceptos como programación evolutiva, simulación, modelos en tiempo de ejecución, entre otros



Take home messages

1. La naturaleza evolutiva innata de los entornos del mundo real ocurre también en sistemas de software modernos
2. Los mecanismos de autoadaptación y autoevolución son habilitadores clave de los sistemas de software inteligentes como los DT
1. Aún existen discontinuidades en el proceso de entrega continua
2. El entorno de ejecución afecta la evolución del software
¡Debemos hacer esto explícito!
3. Estamos trabajando hacia un proceso de evolución en tiempo de ejecución
Ingeniería de Software en tiempo de ejecución

Seminarios 4U

¿PREGUNTAS?

Miguel Jiménez^{1,2}, Luis F. Rivera^{1,2}
miguel@uvic.ca, rivera@uvic.ca

¹Rigi Research, University of Victoria, British Columbia, Canadá
²i2t-DRISO, Universidad Icesi, Cali, Colombia

Investigación patrocinada por



University
of Victoria



UNIVERSIDAD
ICESI



Organizadores



Vigilada Mineducación

